AMI Consortium

http://www.amiproject.org/

Funded under the EU Sixth Framework Programme

Multimodal interfaces action line of the IST Programme

Integrated Projects

AMI (IST-506811) and AMIDA (IST–033812)

State of the Art Report

Meeting Browsing

November 7, 2007

AMI Consortium State of the Art Report

Meeting Browsing

November 7, 2007

Abstract

A meeting browser is an application designed to allow users to access archived meeting recordings. Though browsing might figure heavily when accessing such archives, the application should also support search and any other interactions that could take place between an end user and a meetings archive. This document examines the state of the art of meeting browsers. We begin by re-classifying browsers and related applications into three tiers according to the source of the data they primarily make use of. We look at each tier and discuss the problems faced at each tier and the solutions designed to address these problems. We then examine two browsers in detail - one which offers a complete recording and browsing system, and a meta browser which allows the user to select which components they want to use. We then conclude by briefly examining the process of evaluating meeting browsers.

1 Introduction

Given the ever decreasing cost of capture and storage of multimedia data the recording and archiving of meetings is now relatively common. To access these archives a *meeting browser* is typically used. Despite its name, any application which acts as a front end to a meeting archive is considered to be a meeting browser whether the primary focus is on browsing, search, summarisation, or other forms of interaction. Meeting browsing is an emerging field but despite this there are a large numbers of browsers described in the literature.

To organise meeting browser research this report refines the classification scheme described in Tucker and Whittaker [2005] and Bouamrane and Luz [2007]. There browsers were separated into groups according to the type of data they made primary use of for navigation or presentation. Four groups were selected: audio, video, artefact and discourse browsers (Although Bouamrane and Luz [2007] analyse the first three groups only). In this document we refine this classification by separating browsers into tiers (see Fig. 1). The first tier comprises data *recorded* during the meeting - namely the audio and video recordings. The second tier consists of data that the *participants create* during the meeting - personal notes, slides, minutes etc. The third tier consists of browsers that make primary use of data *derived* from the previous two tiers - speech transcripts, focus of attention, higher level annotations etc. Note that browsers generally also make use of data from lower tiers.

This report continues by examining the requirements of a meeting browser, then analyses meeting browsers that have been developed in the past using this modified taxonomy.We then examine two browsers in detail and conclude by briefly examining how browsers have been evaluated.

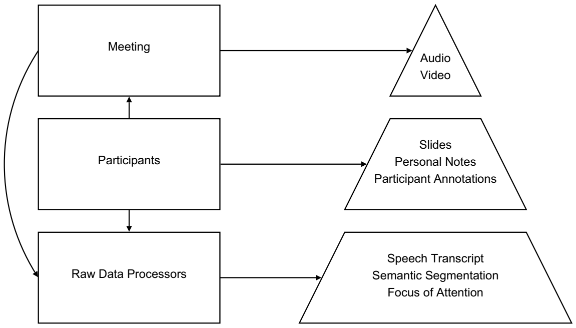


Figure 1: The three tiers of meeting data.

2 Motivations

Little work has examined what users require from meeting browsers but generally one of three methods has been used in order to elicit requirements: Large scale surveys of tools and memory processes, query elicitation and analysis of current practices. These studies are discussed in detail in Whittaker et al. [2008] but we briefly revisit the major findings here.

Jaimes et al. [2004] used a questionnaire to assess the current use of tools to review meetings, finding that meeting participants often reconstructed meeting information rather than remembering it verbatim. A similar approach was used by Lisowska [2003] (and Lisowska et al. [2004a]) who asked participants to generate questions that they would ask of a meeting browser.

Whittaker et al. [1994] examined current recording practices by interviewing people who currently recorded meetings and investigating what the advantages and disadvantages of their recording approaches were. A second interview study investigated the note-taking practices of meeting participants and sought to identify and address the problems that note-taking generates. Additionally, Whittaker et al. [2008] describes an ethnographic study of two firms which investigated the problems of benefits of both personal (notes) and public (minutes) meeting records.

These studies combined have implications for browser designers. Firstly the studies found that users of meeting browsers frequently had the need to access abstractions rather than raw data. Thus, when accessing meeting archives users rarely want to review an entire meeting but instead need to focus on short, relevant parts of the meeting. Secondly users expressed a desire to access meeting archives through categorisations that were relevant to them - e.g. agenda items, decisions and actions. These types of data are often poorly supported by meeting browsers. The overall finding of these studies suggests that browsers are currently too complex and unfocused with regard to user requirements. The two third tier browsers we discuss in detail below go some way towards addressing these criticisms.

3 Meeting Browsers

3.1 First Tier Browsers

Since the first tier browsers are focused on raw multimedia recordings it is easier to examine how the problems of navigating to and locating relevant information have been investigated. Therefore to examine this tier of browsers we look at the solutions to the problems associated with audio rather than at specific applications.

3.1.1 Speech Marking

Degen et al. [1992] exemplifies the *audio marking* approach. Such systems allow users to manually mark relevant points of the meeting as it is recording, here by using a ‘marker’ button on the recorder itself. When playing back the recording the markers can be used as a means of navigating to points of interest. The recording interface has two different marker buttons which can be distinguished when playing back the recording.

3.1.2 Speech Segmentation

As well as allowing users to manually mark (and therefore manually segment) speech recordings other systems have investigated *automatic* segmentation of speech in order to aid navigation. The means by which the recording is segmented varies depending on the application. For example, Hindus and Schmandt [1992] look at segmenting informal workplace discussions according to both person and also by pauses that each person makes between utterances. Other possibilities for segmentation can be more semantic or can make use of prosodic cues (e.g. Arons [1997]) in order to allow the user to navigate the audio recording more efficiently.

3.1.3 Playback mechanisms

Segmenting the recording goes some way to giving users near-random access to speech recordings. However the problem of processing relevant audio still remains - whilst speech is easy to record we are required to listen at around 150 words per minute, whereas we can read at 600 words per minute (and are adept at skimming text to locate regions of interest). Several browsers have examined methods of altering the playback mechanism in order to address this problem.

The primary technique used for achieving this is to speed up the playback rate whilst simultaneously ensuring that the pitch of the speakers remains unchanged. This process is generally implemented using an overlap and add algorithm (e.g. Hejna [1990]) which effectively has a ‘concertina’ like effect on the audio waveform. The concertina effect is inaudible since the ‘folds’ are chosen to align with pitch periods and so the technique is similar to removing a number of pitch vibrations - thus the speech is shortened but the pitch is unchanged. Arons [1997] made use of speed up in the Speech Skimmer device which allowed the user to jointly choose the playback rate and to restrict the playback to relevant segments of the recording (computed according to prosodic cues). Other work has examined the use of speed up (e.g. Tucker and Whittaker [2006a], Arons [1992]) with the general finding that speeds of up three times real time can be understood if enough training is given. However without training sped up speech can sound disconcerting and is a long way from natural speech even with pitch correction.

An alternative technique for allowing listeners to process speech recordings is to use information retrieval and natural language processing algorithms to identify generically 'important' regions of the recording and playback only those regions. This naturally means that the listener is no longer hearing the full recording but the approach has the advantage of not requiring any training since the speech is played back at the natural rate and also that the cognitive limit on the level of compression. Different methods can be used to compute which parts of the recording are important ranging from simple IR metrics (Tucker and Whittaker [2006b]) to more complex summarisation inspired techniques (Murray and Renals [2006]). These two examples make use of speech transcripts to derive the importance scores although it should be noted that this approach to temporal compression of speech need not require transcripts since the importance scores could be computed from purely acoustical measures (Arons [1997]). In addition to this the techniques used are fairly robust to speech transcription errors and it is typically found that the portions of the recording that are scored as having a high importance are well recognised by speech transcription systems (Zechner and Waibel [2000]).

In addition to systems which manipulate the audio stream to allow users to process speech recordings more efficiently there are also approaches which alter the method of *presenting* the audio for the same purpose. An example is Schmandt and Mullins [1995] where different parts of the recording are played simultaneously to both ears. The listener is able to attend to both parts and can focus on parts of the recording that they find interesting. Using the same technology in an alternative way Schmandt [1998] describes an audio playback system where listeners travel down a virtual hallway, hearing snippets of interesting conversation. If the listener identifies a portion of the recording that they are interested in then they can enter the relevant virtual room.

Video recordings have similar problems to those seen for audio recordings. Videos are relatively easy to make and store (although this is more complex than audio) but they are costly to search and browse. Systems that focus on video recordings have largely focused on summarisation and altered playback mechanisms.

3.1.4 Keyframing

One method of overcoming the problem of navigating lengthy video recordings is to represent the video recording as a finite number of *keyframes*. There are a variety of methods for determining which frames should be used as keyframes varying from a random selection to methods which measure the uniqueness of each frame in a series and select the most unique frame to be the designated keyframe.

Typically keyframes are presented in a linear fashion which reflects the temporal evolution of the video. Girgensohm et al. [2001] presented keyframes in a comic book style display by picking keyframes and then measuring the relative importance of each of the keyframes that had been chosen. The keyframes were then laid out in a comic book style where the importance of each keyframe is used to determine how much space the keyframe should take up in the layout. Since the layout is now two dimensional though, the temporal connection is not as clear with this layout as it is with a linear layout.

3.1.5 Video Playback

Other systems have addressed the browsing and access problems by using indexing systems similar to that seen for audio. For example, He et al. [1999] describe a video skimming system which allows users to jump backwards and forwards in time using automatically derived index markers. Additionally, the system allows the user to playback the video at increased speed using the techniques described above for speeding up audio and synchronising the audio with the video.

Foote et al. [1998] also describes a video browser with a variable speed control. Here the user has the option of manually altering the playback rate using a slider mechanism. In addition to manual control the browser also offers an automatic method of varying the playback rate. The playback rate is linked to a confidence measure of 'interestingness' so that the user watches relevant portions of the recording in real time or near real time and portions of the recording which are measured as being uninteresting are played back at a much faster rate.

3.1.6 Video Summarisation

Another means of assisting browsing of digital video is through the creation of skims - an automatically derived multimedia summary of a video (Christel et al. [1998]). The skims here have similarity with the audio skims described above but a key difference is that they not only include a video component but also de-synchronize the audio and video tracks when producing the skim. Thus the skimming process selects the important audio and video sections (which may or may not coincide) and then combine these in a coherent and meaningful way.

3.2 Second Tier Browsers

Second tier browser make use of the interaction technique described in the previous tier but additionally include participant generated data. Therefore the second tier browsers make use of slides, participant notes and any other data that is shown in the meeting or created as a by product of the meeting itself. The type of data produced in meetings can be further categorized into that which is produced by individuals and that which is produced by the community of participants. Artefacts like slides and whiteboard annotations are examples of the latter category and personal notes is a good example of individual data which is produced in a meeting. We examine each of these types of data below.

3.2.1 Slides

Geyer et al. [2001] describes the TeamSpace system which includes elements to support the organisation of meetings as well as providing means to record a meeting and a corresponding interface to review archived meetings. The meeting viewer includes some of the audio segmentation and indexing work described above but centrally focuses on the slides that were presented during the meeting. Thus the user is able to select a slide and listen to audio that was said whilst the slide was being displayed. Additionally the system records any annotations made to the slide.

3.2.2 Whiteboard

The Distributed Meetings client (Cutler et al. [2002]) is another system which integrates several components into a meeting organiser and recorder. The system records video using a panoramic view of the whole meeting room and records audio using a single microphone array to aid localisation and tracking of meeting participants. The system also uses a separate digital camera to capture the whiteboard which has the advantage of capturing who is writing on the board, as well as any non-annotation gestures (such as pointing etc.). The resulting browsing interface shows the whiteboard image centrally, along with audio and video segmentation information. The whiteboard markings are also segmented, again allowing users to select a whiteboard segment and watch the audio and video related to that segment.

Brotherton et al. [1998] describe a system for the visualisation of multiple media streams for the Classroom 2000 project. The purpose of this project is to take a lecture, capture multimedia data from the presentation and then package this data together in a format that supports post hoc browsing and information extraction. The system uses a digital whiteboard to capture annotations during a lecture and then uses post-processing on the whiteboard annotations for segmentation. The level of granularity is much greater here, therefore, since the user is able to select single annotations and determine from this which specific part of audio was being played when this annotation was made. The system also provides a ‘focus of attention’ timeline which indicates at what point during the lecture slides or the whiteboard were the main focus of the class.

3.2.3 Notes

The most common method of integrating personal notes into the browsing interface is by time-stamping each individual ‘note’ and then using this as a supplementary index into the audio and video recordings. This is the approach of the Filochat system (Whittaker et al. [1994]) where a tablet PC was used by individual participants as a means of taking notes during a meeting. The PC also acted as a means of recording the meeting audio. The users of the system could then revisit their notes after the meeting, select a particular annotation and hear the audio that was recorded at the time the note was taken.

A similar approach was taken by Moran et al. [1997], although here the application was designed to be used by a single person rather than by all the participants in a meeting. The chair of the meeting used a PC to write notes in a specific template and then used these notes to revisit the meeting and make a particular decision. Both these studies found that in addition to supporting note taking practices the introduction of these system lead to changes in the way that participants take notes. Specifically, Moran et al. [1997] noted that users of the system would often make short notes (“ha” was used for this) to indicate something that was interesting and that should be re-listened to later. Chiu et al. [1999] also implements this paradigm but extends it to account for multimedia. Here users are able to annotate chosen video frames or presented slides as they wish. They can also select an automatic setting where new slides or significant changes to the video add a new pane to the display onto which the user can make notes. The user can then review the recording by looking at the static slide and video captures.

3.2.4 Minutes

The final area of participant created data are the meeting minutes. To support the taking of minutes Chiu et al. [2001] allowed the designated scribe to take the minutes on a wirelessly connected laptop. Whilst the minutes are being taken an audio, video and slide capture recording is made of the meeting. At the end of the meeting the minutes are then distributed in a variety of formats, some of which contain links back to the slides and video of relevant sections of the meeting. The minuting system also allows participants to revise the minutes as necessary, with the revisions being passed on to the other meeting participants.

The MinuteAid (Lee et al. [2004]) system takes a similar approach. The system here differs in that it allows the scribe to take minutes during the meeting and add any multimedia created during the meetings (slides, video frames) as they desire during the meeting. Thus the resulting meeting minutes are manually constructed but with user specified references to the multimedia content.

3.3 Third Tier Browsers

The third tier of meeting browsers have access to the views and data provided by the first and second tiers but embellish these with data derived from the raw meeting content and different presentations of this data. Whereas most of the browsers above were focused on single types of data or presentations the browsers in this tier tend to take a broad view and, in some cases, could be considered fully featured state of the art meeting browsers.

Whittaker et al. [2002] outlines the ScanMail system which, although is designed to work with voicemail, has functionality which could be applied to meetings. In ScanMail voicemail messages are converted into enhanced emails using a combination of speech transcription and post processing. When a voicemail is left for the user the system converts it into text using an automatic speech transcriber. The user then has a perspective on their voicemail which is like an email reader. The speech and text is synchronised so that the user is able to listen to specific parts of the voicemail by selecting the relevant parts of the text. Thus if there are any sections of the transcript which appear to contain transcription errors the user can immediately verify what the correct text should be by listening to the corresponding portion of audio. The system also allows user to search their voicemail with a text search and marks up parts of the message, such as phone numbers, so that users can easily extract important information from the message.

The Rough N Ready browser (Colbath et al. [2000]), like ScanMail, is not focused on the meeting domain but again contains ideas which are applicable to meeting browsers. The system starts by processing news recordings and transcribing the speech. Again the transcription links back to the audio recording so the user can choose to listen to a portion of the audio recording at any time by selecting a part of the transcript. Additionally users are able to search for specific entities, such as people, locations, organizations. Search results can be displayed on a timeline indicating the temporal density of the search results.

A meeting browser which gives the transcript prominence is described in Bett et al. [2000]. Here the interface consists of the transcript and a single video component alongside a list of participants and a timeline indicating when each of the participants was speaking. In addition to this archival browser the system allows the user to construct summaries using audio, video and text of the whole meeting or specified parts of the meeting. The interface also allows for the display of various discourse features in a browser and also allows users to search the entire meeting archive.

Lisowska et al. [2004b] describe ARCHIVUS, a system designed to allow users to browse and access multimodal meetings through search or by browsing. The system uses a library metaphor in its interface. Thus each meeting is represented as a book on a shelf - opening a book from the shelf reveals the transcript of the meeting. In addition to the textual transcript the user has access to multimedia elements related to the meeting. In this system the user also has the choice of accessing the archive through speech.

A browser which also examined search is the Transcript-Based Query and browsing interface (TQB) described in Popescu-Belis and Georgescul [2006]. The TQB interface allows the user to enter free text queries and search over a set of meeting transcripts for utterances which contain these queries. The interface also allows user to focus their searches, for example by searching for utterances by a single participant or utterances which are of a particular type (e.g. a question). The TQB also allows the user to browse the meeting archive by selecting particular episodes or keywords of a particular meeting or by jumping into points where certain documents were discussed. ARCHIVUS and TQB were developed as part of the IM2 project which also developed a number of different meeting browsers which are described in detail in Lalanne et al. [2005].

Jabber (Kazman et al. [1996]) and Ferret (Wellner et al. [2004]) take a similar approach to visualising a recorded meeting. Both focus on a temporal view showing which participants spoke at which point of the meeting. Both contain video components, with Ferret allowing for multiple video components showing different views of the meeting. In addition to these components the Jabber browser shows an overview of the meeting by plotting a graph of involvement for each participant over the course of the whole meeting

The final browser in this category is the document focused browser (Lalanne et al. [2003]). Again this browser displays a transcript and segments the audio into discrete meeting sections. The browser also shows the participant involvement but uses a circular representation to show this alongside several other types of temporal metadata. The focus of this browser is, however, a document and the browser is able to highlight parts of the document that are currently being discussed. Thus the user is able to select a part of the document and then hear the audio that relates to that particular section. This notion has also been extended to look at relationships between discrete multimedia elements (Lalanne et al. [2007]).

4 State of the Art Browsers

We now compare two state of the art browsers. JFerret is an extension of the Ferret browser described above, and the Portable Meeting Recorder (Lee et al. [2002]) is a portable meeting recording and browsing system. Both are interesting because although they fall into the third tier of browsing they do more than just visualising raw data streams and offer a flavour of a possible fourth tier of browsing.

The portable meeting browser (see Fig. 3) encompasses all stages of the meeting capture process. The processing begins with a small single camera which uses a parabolic mirror to enable it to capture a full 360° display of the meeting. At the base of the camera are four microphones placed in a square formation to allow for beam forming which used in later stages of processing. Thus the recordings made are just audio and video and the recording interface allows users to make annotations as the meeting is recorded but this seems to be intended for users of the system rather than for making personal notes.

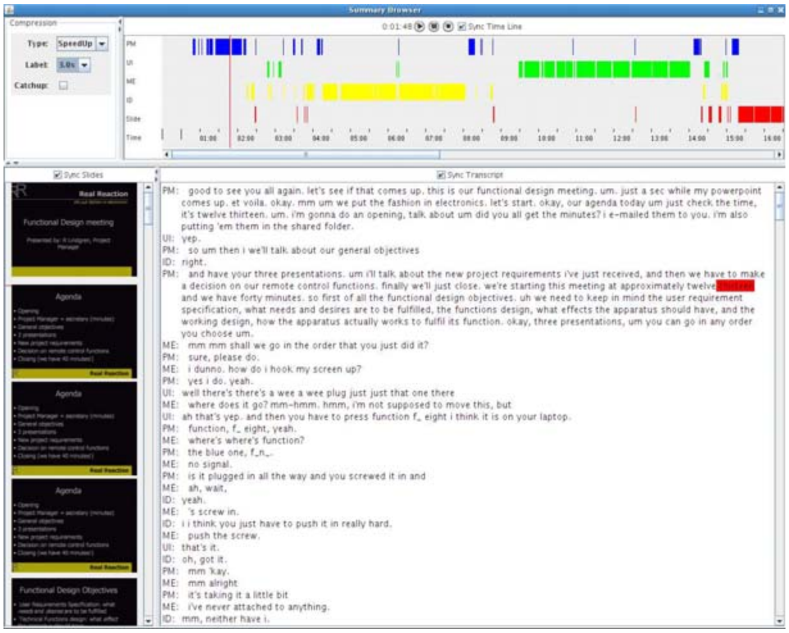
Following the meeting, five sets of post processing are carried out to produce meeting metadata. Firstly the audio streams are processed in order to localize each speaker in the meeting room, here the azimuth angles of each speaker is computed relative to the recording device. On the basis of this an algorithm produces a single video stream which automatically determines the best view for the meeting recording. The current speaker location and a measure of visual change is used to identify the best frame in each case. The background image is then extracted and matched against a database of room templates in order to identify the meeting location.

Once these automatic annotations have been made and placed into the metadata database, the meeting description document is produced which contains all the information that has been automatically extracted - the meeting date, time, location and participants along with an image of the participant. The user can then access the archived recording and search and automatically produced speech transcription to locate points of interest and then watch the corresponding portions of the meeting. In addition to the text search the interface offers the option for users to scan a graph of audio and visual activity in order to locate points of interest.

All of these data are then placed into a rich user interface which includes the activity indexes, speaker participation, key frames of the video, the chosen best shot, the overall panoramic view and the meeting overview. Users can jump to any point in the meeting by selecting from these components

JFerret (see Fig. 2) differs from the browsers above (including the Ferret browser) in that it is a *framework* for building meeting browsers (e.g. Murray et al. [2007]). JFerret allows browser designers to not only layout the various components in a way that suits the intended application but also the components themselves can be altered and chosen in order to make the most efficient browser for the given application. The user of JFerret selects which components they wish to include in their browser and uses a simple XML file to specify where these components should be placed in the screen display. The framework ensures offers a central point of synchronisation which ensures that changes made in one component are reflected in all of the other components in the display.

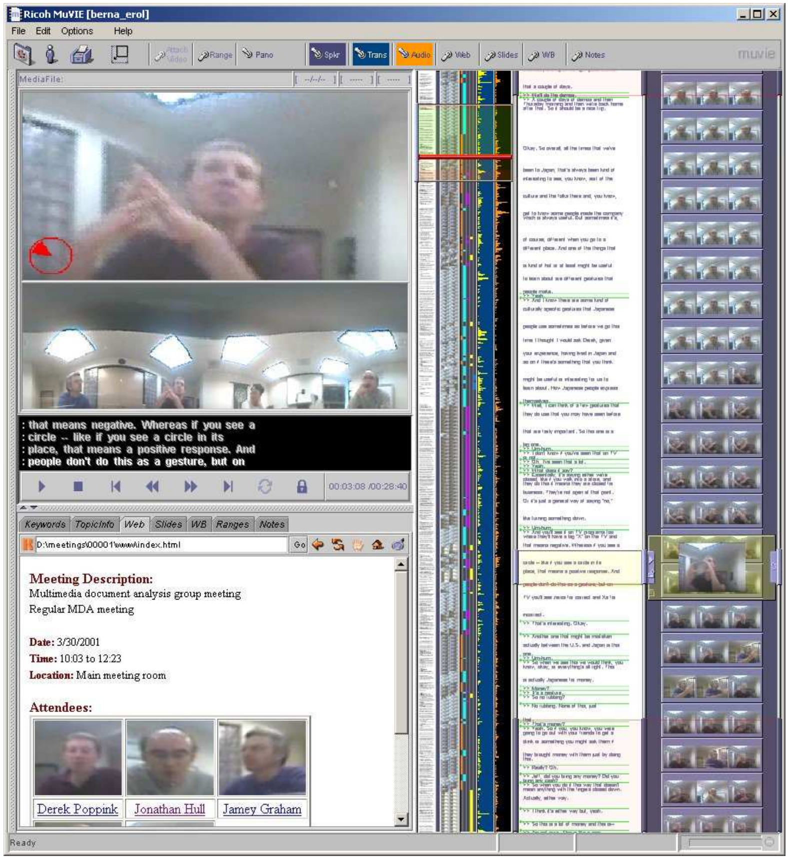
Thus the expressive power of JFerret can be found in the superset of all the components that are available to the browser designer. In keeping with the browsers described above JFerret offers a component for displaying video and playing back audio. The system also is able to show slides, any kind of speech transcript, and the personal notes of each of the meeting participants. In addition to these more basic components JFerret includes components for playing back audio at faster rates using the overlap and add techniques described above and more semantically motivated playback techniques (Tucker and Whittaker [2006a]). The browser can also include a summarisation component consisting of



Type: SpeedUp
Label: 1.0x
Catchup: □

PM: good to see you all again. let's see if that comes up. this is our functional design meeting. um. just a sec while my powerpoint comes up. et voila. okay. mm um we put the fashion in electronics. let's start. okay, our agenda today um just check the time, it's twelve thirteen. um. i'm gonna do an opening, talk about um did you all get the minutes? i e-mailed them to you. i'm also putting 'em them in the shared folder.
UI: yep.
PM: so um then i we'll talk about our general objectives
ID: right.
PM: and have your three presentations. um i'll talk about the new project requirements i've just received, and then we have to make a decision on our remote control functions. finally we'll just close. we're starting this meeting at approximately twelve thirteen and we have forty minutes. so first of all the functional design objectives. uh we need to keep in mind the user requirement specification, what needs and desires are to be fulfilled, the functions design, what effects the apparatus should have, and the working design, how the apparatus actually works to fulfil its function. okay, three presentations, um you can go in any order you choose um.
ME: mm mm shall we go in the order that you just did it?
PM: sure, please do.
ME: i dunno. how do i hook my screen up?
PM: yes i do. yeah.
UI: well there's there's a wee a wee plug just just that one there
ME: where does it go? mm-hmm. hmm, i'm not supposed to move this, but
UI: ah that's yep. and then you have to press function f_ eight i think it is on your laptop.
PM: function, f_ eight, yeah.
ME: where's where's function?
PM: the blue one, f_n_.
ME: no signal.
PM: is it plugged in all the way and you screwed it in and
ME: ah, wait,
ID: yeah.
ME: 's screw in.
ID: i i think you just have to push it in really hard.
ME: push the screw.
UI: that's it.
ID: oh, got it.
PM: mm 'kay.
ME: mm alright
PM: it's taking it a little bit
ME: i've never attached to anything.
ID: mm, neither have i.

Figure 2: The JFerret Meeting Browser showing a time line, compressed audio player, meeting slides and transcript.



: that means negative. Whereas if you see a
: circle -- like if you see a circle in its
: place, that means a positive response. And
: people don't do this as a gesture, but on

00:03:08 /00:28:40

D:\meetings\00001\www\index.html

Meeting Description:
Multimedia document analysis group meeting
Regular MDA meeting

Date: 3/30/2001
Time: 10:03 to 12:23
Location: Main meeting room

Attendees:

Derek Poppink
Jonathan Hull
Jamey Graham

Figure 3: The Portable Meeting Browser showing videos, keyframes, transcripts, overviews and activity graphs.

both extractive and abstractive summaries which are linked back into the transcript so that the user can select a part of the summary and see where in the transcript the phrases originated.

The display can also include components which indicate the dominance of each speaker (Rienks [2007]), allowing the user of the browser to identify who is the dominant speaker at any point during the meeting. In addition to this there is also a component which allows the user to annotate and track the strands of various arguments throughout the meeting. Arguments are shown in a tree like structure showing each thread of the argument and when parties are in disagreement.

JFerret also allows the user to directly search the ASR output in order to locate keywords (Szoke et al. [2005]). This search is carried out on the ASR lattice and is, therefore, more powerful than doing a text search of the meeting transcript as potential candidates for the keyword match which were rejected by the ASR algorithm can be included in the search. In addition to this users are also able to do a simple text search over the meeting transcript. In addition to these components JFerret also includes a device to allow people to share three dimensional data models and manipulate them whilst in a meeting.

5 Evaluation

Given that meeting browser is still an emergent field it is unsurprising that little work has addressed the evaluation of the efficiency of the meeting browser (see Cremers et al. [2006] for a summary of AMI work). However, now that standard corpora exist Carletta [2006] future work will address this problem. Two relatively large scale evaluations have taken place which suggest directions that future evaluations will take.

The Meeting Browser Evaluation Test (BET) takes a TREC[1] like approach to evaluating meeting browsers. There is a lengthy one-time data collection process and then a relatively rapid evaluation between subjects evaluation phase. The advantage of this process is that the results of the data collection phase can be shared between evaluators and so evaluations can take place in different locations and at different times.

The core of the BET is the notion of *observations of interest*. An observation of interest is pair of statements (one true and one false) which addresses a singular fact about the meeting. Thus an observation may be "The budget was 100 pounds" paired with "The budget was 300 pounds". In the data collection phase a small number of judges watch a meeting in its entirety and then re-watch the meeting and generate these observations of interest. The collection of observations of interest form the basis of the test set which can be shared between evaluators. In the evaluation phase, users are given a novel meeting browser and asked to validate as many observations of interest as they can - browser efficiency is then measured as the number of observations that can be validated in a given amount of time. The experimenters can also log the media time points that the observations were answered at given an indication of whether the user was guessing the answer or actually spotted it in the meeting. The BET framework has also been extended to allow experimenters to have more control over what kind of observations are used when evaluating browsers and this

[1] http://trec.nist.gov/

extended form of the BET has been used for evaluation campaigns Popescu-Belis et al. [2007].

It could be argued that the BET is an intrinsic evaluation - it measures browser performance in a simulated task that approximates one use of the browser in the real world. An example of an extrinsic evaluation can be found in Elling [2007]. Here browser performance is measured by adding the technology into a simulated meeting and seeing how the meeting process is improved as a result of supplying different browser systems to the team. The users are told that they are replacing a team who have previously met regarding a project to build a new remote control for the television. The previous meetings were recorded and the new team are provided with different meeting browsers in order to review the prior work. A large number of performance measures are then used to measure how successful each team is.

The extrinsic evaluation has the advantage of placing the users in a more realistic environment where the performance of the browser is critical to their success. The drawback to this is that the experimenter is unable to make fine-grained assessments of the browser performance - it is difficult to draw out why a particular browser worked well or what specific questions the browser was adept at answering. These types of questions could be answered by a BET style evaluation approach.

6 Conclusion

We have shown how meeting browsers can be considered to be in one of three tiers depending on the type of data that the browser focuses on. First tier meeting browser make direct use of the raw data streams that are recorded during a meeting - thus they concentrate on the audio or video. Second tier browsers make use of and focus on the data that the meeting participants create or present during the meeting - slides, minutes, personal notes etc. Third tier data is that derived from the raw and participant data namely things like ASR transcripts, participant involvements, locations etc. Browsers in each tier generally make use of principles and data from the tiers above it so that third tier browsers can be considered state of the art browsers. We also examined two third tier browsers in detail - one which was a complete meeting recording and browsing solution and one which is effectively a framework for combining browser components into a single browser. We also briefly examined how such browsers are evaluated.

References

B. Arons. Techniques, perception, and applications of time-compressed speech. In *1992 Conference, American Voice I/O Society*, pages 169–177, September 1992.

B. Arons. Speechskimmer: A system for interactively skimming recorded speech. *ACM Transcations on Computer-Human Interaction*, 4(1):3–38, March 1997.

M. Bett, R. Gross, H. Yu, X. Zhu, Y. Pan, J. Yang, and A. Waibel. Multimodal meeting tracker. In *RIAO*, April 2000.

M. Bouamrane and S. Luz. Meeting browsing: state-of-the-art review. *Mulitmedia Systems*, 12:439–457, 2007.

J. A. Brotherton, J. R. Bhalodia, and G. D. Abowd. Automated capture, integration and visualization of multiple media streams. In *The IEEE International Conference on Multimedia Computing And Systems*, pages 54–63, 1998.

J. Carletta. Announcing the AMI meeting corpus. *The ELRA Newsletter*, 11(1):3–5, 2006.

P. Chiu, A. Kapuskar, S. Reitmeier, and L. Wilcox. Notelook: Taking notes in meetings with digital video and ink. In *ACM Multimedia '99*, 1999.

P. Chiu, J. Boreczky, A. Girgensohn, and D. Kimber. Liteminutes: An internet-based system for multimedia meeting minutes. In *10th WWW Conference*, pages 140–149, May 2001.

M.G. Christel, M.A. Smith, C. Roy Taylor, and D.B. Winkler. Evolving video skims into useful multimedia abstractions. In *CHI '98*, April 1998.

S. Colbath, F. Kubala, D. Liu, and A. Srivastava. Spoken documents: Creating searchable archives from continuous audio. In *33rd Hawaii International Conference On System Sciences*, 2000.

A. Cremers, W. Post, E. Elling, B. van Dijk, B. van derWal, J. Carletta, M. Flynn, P. Wellner, and S. Tucker. Meeting browser evaluation report. Technical report, AMI Project Deliverable, 2006.

R. Cutler, Y. Rui, A. Gupta, J.J. Cadiz, I. Tashev, L. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: A meeting capture and broadcasting system. In *10th ACM International Conference on Multimedia*, pages 503–512, December 2002.

L. Degen, R. Mander, and G. Salomon. Working with audio: Integrating personal tape recorders and desktop computers. In *CHI '92*, pages 413–418, May 1992.

E. Elling. Tools for fun and fruitful meetings. Master's thesis, University of Twente, 2007.

J. Foote, J. Boreczky, A. Girgensohn, and L. Wilcox. An intelligent media browser using automatic multimodal analysis. In *ACM Multimedia*, pages 375–380, September 1998.

W. Geyer, H. Richter, L. Fuchs, T. Frauenhofer, S. Daijavad, and S. Poltrock. A team collaboration space supporting capture and access of virtual meetings. In *2001 International ACM SIGGROUP Conference On Supporting Group Work*, pages 188–196, September-October 2001.

A. Girgensohm, J. Borczky, and L. Wilcox. Keyframe-based user interfaces for digital video. *IEEE Computer*, 34(9):61–67, September 2001.

L. He, E. Sanocki, A. Gupta, and J. Grudin. Auto-summarization of audio-video presentations. In *7th ACM International Conference On Multimedia*, pages 489–498, 1999.

D.J. Hejna. Real-time time-scale modification of speech via the synchronized overlap-add algorithm. Master's thesis, M.I.T., 1990.

D. Hindus and C. Schmandt. Ubiquitous audio: Capturing spontaneous collaboration. In *1992 ACM Conference on Computer-Supported Cooperative Work*, pages 210–217, November 1992.

A. Jaimes, K. Omura, T. Nagamine, and K. Hirata. Memory cues for meeting video retrieval. In *CARPE '04*, pages 74–85, October 2004.

R. Kazman, R. Al-Halimi, W. Hunt, and M. Mantei. Four paradigms for indexing video conferences. *IEEE Multimedia*, 3(1):63–73, Spring 1996.

D. Lalanne, S. Sire, R. Ingold, A. Behera, D. Mekhaldi, and D. Rotz. A research agenda for assessing the utility of document annotations in multimedia databases of meeting recordings. In *3rd International Workshop on Multimedia Data And Document Engineering*, September 8th 2003.

D. Lalanne, A. Lisowska, E. Bruno, M. Flynn, M. Gerogescul, M. Guillemot, B. Janvier, S. Marchand-Maillet, M. Melichar, N. Noenne-Loccoz, A. Popescu-Belis, M. Rajman, M. Rigamonti, D. Rotz, and P. Wellner. The IM2 multimodal meeting browser family. Technical report, IM2 Project, 2005.

D. Lalanne, M. Rigamonti, F. Evequoz, B. Dumas, and R. Ingold. An ego-centric and tangible approach to meeting indexing and browsing. In Bourlard H. & Renals S. Popescu-Belis A., editor, *Machine Learning for Multimodal Interaction IV, Revised Selected Papers, LNCS*. Springer-Verlag, Berlin/Heidelberg, 2007.

D. Lee, J.J. Hull, B. Erol, and J. Graham. Minuteaid: Multimedia note-taking in an intelligent meeting room. In *IEEE International Conference on Multimedia and Expo*, 2004.

D-S Lee, B. Erol, J. Graham, J. J. Hull, and N. Murata. Portable meeting recorder. In *ACM Multimedia*, pages 493–502, 2002.

A. Lisowska. Multimodal interface design for the multimodal meeting domain: Preliminary indications from a query analysis study. Technical Report IM2.MDM Report 11, IM2, November 2003.

A. Lisowska, A. Popescu-Belis, and S. Armstrong. User query analysis for the specification and evaluation of a dialogue processing and retreival system. In *Proceedings of LREC 2004*, volume III, pages 993–996, 2004a.

A. Lisowska, M. Rajman, and T.H. Bui. Archivus: A system for accessing the content of recorded multimodal meetings. In *MLMI 2004*, 2004b.

T.P. Moran, L. Palen, S. Harrison, P. Chiu, D. Kimber, S. Minneman, W. Melle, and P. Zellweger. ”i’ll get that off the audio”: A case study of salvaging multimedia meeting records. In *CHI ’97*, 22-27 March 1997.

G. Murray and S. Renals. Dialogue act compression via pitch contour preservation. In *Proceedings of the 9th International Conference on Spoken Language Processing, Pittsburgh, USA*, September 2006.

G. Murray, P. Hsueh, S. Tucker, J. Kilgour, J. Carletta, J.D. Moore, and S. Renals. Automatic segmentation and summarization of meeting speech. In *Proceedings of NAACL-HLT 2007*, April 2007.

A. Popescu-Belis and M. Georgescul. Tqb: Accessing multimedia data using a transcript-based query and browsing interface. In *Proceedings of LREC 2006*, pages 1560–1565, 2006.

A. Popescu-Belis, P. Baudrion, M. Flynn, and P. Wellner. Towards an objective test for meeting browsers: the BET4TQB pilot experiment. In Bourlard H. & Renals S. Popescu-Belis A., editor, *Machine Learning for Multimodal Interaction IV*, pages 108–119. Springer-Verlag, Berlin/Heidelberg, 2007.

R. Rienks. *Meetings in smart enviornments: Implications of progressing technology*. PhD thesis, University of Twente, 2007.

C. Schmandt. Audio hallway: A virtual acoustic environment for browsing. In *UIST*, pages 163–170, 1998.

C. Schmandt and A. Mullins. Audiostreamer: Exploting simultaneity for listening. *Proceedings of CHI '95*, 1995.

I. Szoke, P. Schwarz, P. Matejka, L. Burget, M. Karafiat, and J. Cernocky. Phoneme based acoustics keyword spotting in informatl continuous speech. In V. Matousek, editor, *Lecture Notes In Computer Science*, volume 2658, pages 302–309. Springer-Verlag, 2005.

S. Tucker and S. Whittaker. Accessing multimodal meeting data: Systems, problems and possibilities. In S. Bengio and H. Bourlard, editors, *Lecture Notes In Computer Science*, volume 3361, pages 1–11. Springer-Verlag, 2005.

S. Tucker and S. Whittaker. Displaying dynamic meeting transcripts: Concertina browsing. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, May 2006a.

S. Tucker and S. Whittaker. Time is of the essence: An evaluation of temporal compression algorithms. In *Proceedings of CHI '06*, April 2006b.

P. Wellner, M. Flynn, and M. Guillemot. Browsing recording of multi-party interactions in ambient intelligent envrionments. In *CHI*, April 2004.

S. Whittaker, P. Hyland, and M. Wiley. Filochat: Handwritten notes provide access to recorded conversations. In *Chi '94*, 271-277, April 1994.

S. Whittaker, J. Hirschberg, B. Amento, L. Stark, M. Bacchiani, P. Isenhour, L. Stead, G. Zamchick, and A. Rosenberg. SCANMail: A voicemail interface that makes speech browsable, readable and searchable. In *Proceedings of CHI 2002*, April 2002.

S. Whittaker, S. Tucker, K. Swampillai, and R. Laban. Design and evaluation of systems to support interaction capture and retrieval. *Personal and Ubiquitous Computing*, 2008. In press.

K. Zechner and A. Waibel. Minimizing word error rate in textual summaries of spoken language. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics, NAACL-200*, pages 186–193, Seattle, WA., April / May 2000.